TASK-1: Prediction using Supervised ML - Spark Foundation

Manthan Bhikadiya

Import Dependencies

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
data = pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/studen
```

In [3]:

```
data.head()
```

Out[3]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [4]:

```
len(data)
```

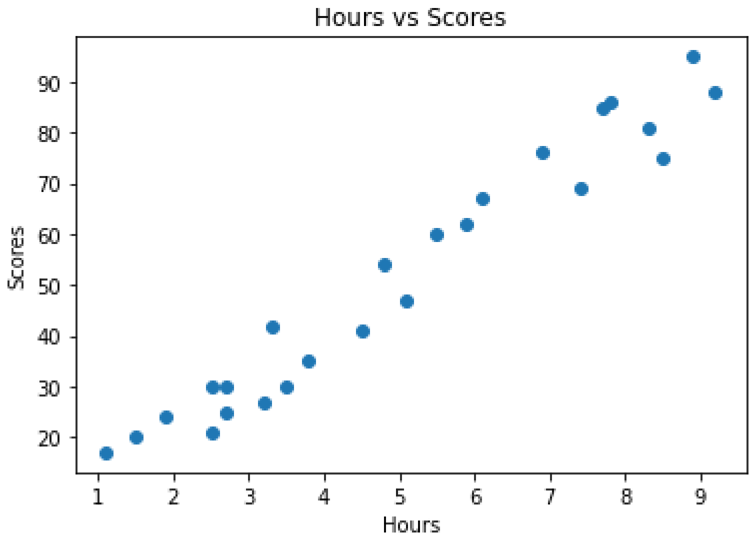
Out[4]:

```
25
```

Data Distribution

In [5]:

```
plt.scatter(x = 'Hours' ,y = 'Scores' , data = data)
plt.title('Hours vs Scores')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show()
```



In [6]:

```
features = data.iloc[: , 0]
target = data.iloc[: , 1]
```

Train Test Split

In [7]:

```
from sklearn.model_selection import train_test_split
```

In [8]:

```
X_train,X_test , y_train , y_test = train_test_split(features , target , test_size = 0.2 ,
```

In [9]:

```
from sklearn.linear_model import LinearRegression
```

Linear Regression

In [10]:

```
lr = LinearRegression()
```

In [11]:

```
X_train = X_train.values.reshape((-1,1))
```

In [12]:

```
X_test = X_test.values.reshape((-1,1))
```

In [13]:

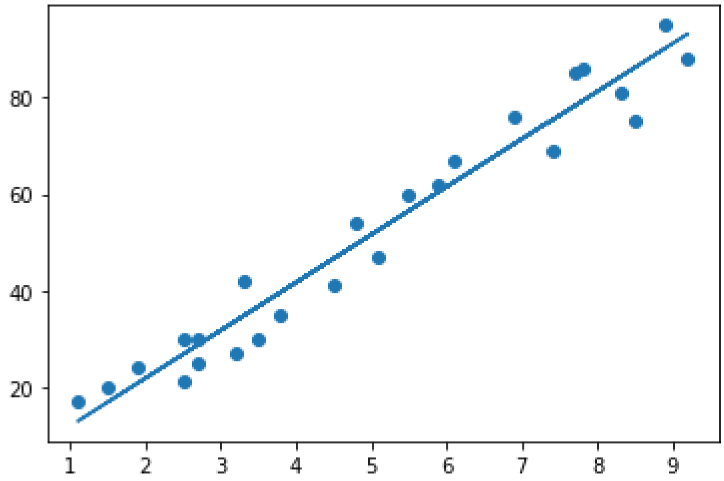
```
lr.fit(X_train , y_train )
```

Out[13]:

```
LinearRegression()
```

In [14]:

```
y = lr.coef_*features + lr.intercept_
plt.scatter(x = 'Hours' ,y = 'Scores' , data = data)
plt.plot(features , y )
plt.show()
```



Predictions

In [15]:

```
predictions = lr.predict(X_test)
```

In [16]:

```
pd.DataFrame({'Original': y_test ,'Predicted':predictions })
```

Out[16]:

	Original	Predicted
5	20	16.884145
2	27	33.732261
19	69	75.357018
16	30	26.794801
11	62	60.491033

What will be predicted score if a student studies for 9.25 hrs/ day?

In [17]:

```
hours = 9.5
answer = lr.predict([[hours]])
print(f"Predicted score :{answer[0]:.3f} if a student studies for 9.25 hrs/ day")
```

```
Predicted score :96.169 if a student studies for 9.25 hrs/ day
```

Model Metrics

In [18]:

```
from sklearn.metrics import mean_squared_error, r2_score
```

In [19]:

```
print(f"Mean Squared Error : {mean_squared_error(y_test , predictions)}")
print(f"Root Mean Squared Error : {mean_squared_error(y_test , predictions , squared = Fals
print(f"R2-Score : {r2_score(y_test , predictions)}")
```

```
Mean Squared Error : 21.5987693072174
Root Mean Squared Error : 4.6474476121003665
R2-Score : 0.9454906892105356
```